# Наследвание в классах

class A {  
 String version = "1.0 A";  
  
 String testMetod() {  
 return "A";  
 }  
}  
  
class B extends A {  
 String version = "2.0 B";  
 String testMetod(){  
 return "B";  
 }  
 }  
public class Main2 {  
 public static void main(String[] args) {  
 A a = new B();  
 System.*out*.println(((B) a).version + a.testMetod());  
 }  
}

вывод

2.0 BB

при

System.*out*.println(a.version + a.testMetod());

Вывод

1. AB

# Перегрузка метода

Это статический полиморфизм, т.к. версия метода определяется на этапе компиляции.

public void method() {}  
public void method(int value) {}

в программе

public class Point2D extends Point1D{  
 private int y;  
  
 public int getY() {  
 return y;  
 }  
  
 public void setY(int y) {  
 this.y = y;  
 }  
  
 @Override  
 public double length() {  
 return Math.*hypot*(y, getX());  
 }  
 public void method() {}  
 public void method(int value) {}  
}

Вызов статического метода

public class LearnMainOverload {  
 public static void main(String[] args) {

Point2D point = new Point2D();

Point2D.*method*("java");//Вызов статического метода

}

Его формирование

public class Point2D extends Point1D{  
 private int y;  
  
 public int getY() {  
 return y;  
 }  
  
 public void setY(int y) {  
 this.y = y;  
 }  
  
 @Override

public static void method(String str) {  
 System.*out*.println("String " + str);  
}//Перегрузка метода

Вызов метода

public class LearnMainOverload {  
 public static void main(String[] args) {  
 Point2D point = new Point2D();  
 point.method();}

Формирование метода

@Override

public void method() {  
 System.*out*.println("without");  
}

1. Не использовать перегрузку с одинаковым числом параметров.
2. Не использовать сложных вариантов перегрузки
3. И лучше перегрузку заменять различными методами 😊

# Перегрузка конструктора

Конструктор – это специальный метод класса, предназначение которого инициализация полей класса. У любого класса есть как минимум один конструктор.

Конструктор по умолчанию

public class Person {  
 private int id;  
 private String name = "";  
 public Person(){  
 super();//Конструктор

}  
}

Перегрузка конструктора

public class Person {  
 private int id;  
 private String name = "";  
 public Person(){  
 super();//Конструктор  
 }  
  
 public Person(int id) {//Перегрузка конструктора  
 this.id = id;  
 }  
 public Person(String name) {  
 this.name = name;  
 }  
  
 public Person(int id, String name) {4  
 this.id = id;  
 this.name = name;  
 }  
}

При

public Person(int id, String name) {4  
 this.id = id;  
 this.name = name;  
 }

Конструктор по умолчанию не предостовляется.

Арт. 82737124

# **Настольная лампа светодиодная Uniel TLD-546, нейтральный белый свет, цвет белый**

Арт. 82810202

# **Торшер Мотив, цвет белый/хром**